

Metodos de Strings en Python

1. capitalize()

- Descripción: Convierte la primera letra de cada palabra en mayúscula y las demás letras en minúsculas.
- Ejemplo:

python

```
s = "hello world"
print(s.capitalize()) # Output: "Hello world"
```

2. casefold()

- Descripción: Convierte el string a minúsculas y elimina los caracteres no ASCII.
- Ejemplo:

python

```
s = "Hello, World!"
print(s.casefold()) # Output: "hello, world!"
```

3. center(width, fillchar)

- Descripción: Centra el string en un campo de ancho width y rellena con el carácter fillchar.
- Ejemplo:

python

```
s = "hello"
print(s.center(10, "*")) # Output: "****hello****"
```

4. count(substring)

- Descripción: Devuelve el número de veces que el substring substring se encuentra en el string.
- Ejemplo:

python

```
s = "hello world hello"
print(s.count("hello")) # Output: 2
```

5. encode(encoding, errors)

- Descripción: Codifica el string según el esquema de codificación encoding y maneja errores según errors.
- Ejemplo:

python

```
s = "hello"
print(s.encode("utf-8", "replace")) # Output: b'hello'
```

6. endswith(suffix)

- Descripción: Verifica si el string termina con el sufijo suffix.
- Ejemplo:

python

```
s = "hello world"
print(s.endswith("world")) # Output: True
```

7. expandtabs(tabsize)

- Descripción: Reemplaza los caracteres de tabulación (\t) con espacios y ajusta la cantidad de espacios según tabsize.
- Ejemplo:

python

```
s = "hello\tworld"
print(s.expandtabs(4)) # Output: "hello    world"
```

8. find(substring)

- Descripción: Devuelve la posición de la primera aparición del substring substring en el string. Si no se encuentra, devuelve -1.
- Ejemplo:

python

```
s = "hello world"
print(s.find("world")) # Output: 6
```

9. format(*args, **kwargs)

- Descripción: Formatea el string según los argumentos args y kwargs.
- Ejemplo:

python

```
s = "Hello, {}!"  
print(s.format("world")) # Output: "Hello, world!"
```

10. format_map(mapping)

- Descripción: Formatea el string según el diccionario mapping.
- Ejemplo:

python

```
s = "Hello, {name}!"  
print(s.format_map({"name": "world"})) # Output: "Hello,  
world!"
```

11. index(substring)

- Descripción: Devuelve la posición de la primera aparición del substring substring en el string. Si no se encuentra, lanza una excepción.
- Ejemplo:

python

```
s = "hello world"  
print(s.index("world")) # Output: 6
```

12. isalnum()

- Descripción: Verifica si el string solo contiene caracteres alfanuméricos.
- Ejemplo:

python

```
s = "hello123"  
print(s.isalnum()) # Output: True
```

13. isalpha()

- Descripción: Verifica si el string solo contiene caracteres alfabéticos.
- Ejemplo:

python

```
s = "hello"  
print(s.isalpha()) # Output: True
```

14. isdecimal()

- Descripción: Verifica si el string solo contiene caracteres decimales.
- Ejemplo:

python

```
s = "123"  
print(s.isdecimal()) # Output: True
```

15. isdigit()

- Descripción: Verifica si el string solo contiene caracteres numéricos.
- Ejemplo:

python

```
s = "123"  
print(s.isdigit()) # Output: True
```

16. isidentifier()

- Descripción: Verifica si el string es un identificador válido (nombre de variable, etc.).
- Ejemplo:

python

```
s = "hello"  
print(s.isidentifier()) # Output: True
```

17. islower()

- Descripción: Verifica si el string solo contiene letras minúsculas.
- Ejemplo:

python

```
s = "hello"  
print(s.islower()) # Output: True
```

18. isnumeric()

- Descripción: Verifica si el string solo contiene caracteres numéricos.
- Ejemplo:

python

```
s = "123"  
print(s.isnumeric()) # Output: True
```

19. isprintable()

- Descripción: Verifica si el string solo contiene caracteres imprimibles.
- Ejemplo:

python

```
s = "hello"  
print(s.isprintable()) # Output: True
```

20. isspace()

- Descripción: Verifica si el string solo contiene espacios en blanco.
- Ejemplo:

python

```
s = "   "  
print(s.isspace()) # Output: True
```

21. istitle()

- Descripción: Verifica si el string sigue el formato de título (primera letra de cada palabra en mayúscula).
- Ejemplo:

python

```
s = "Hello World"  
print(s.istitle()) # Output: True
```

22. isupper()

- Descripción: Verifica si el string solo contiene letras mayúsculas.
- Ejemplo:

python

```
s = "HELLO"  
print(s.isupper()) # Output: True
```

23. join(iterable)

- Descripción: Une los elementos de la secuencia `iterable` con el string.
- Ejemplo:

python

```
s = "-"  
print(s.join(["hello", "world"])) # Output: "hello-world"
```

24. ljust(width, fillchar)

- Descripción: Alinea el string a la izquierda en un campo de ancho `width` y rellena con el carácter `fillchar`.
- Ejemplo:

python

```
s = "hello"  
print(s.ljust(10, "*")) # Output: "hello*****"
```

25. lower()

- Descripción: Convierte el string a minúsculas.
- Ejemplo:

python

```
s = "HELLO"  
print(s.lower()) # Output: "hello"
```

26. lstrip(chars)

- Descripción: Elimina los caracteres de la cadena `chars` de la izquierda del string.
- Ejemplo:

python

```
s = "  hello  "  
print(s.lstrip()) # Output: "hello  "
```

27. maketrans(from, to)

- Descripción: Crea un diccionario de traducción para reemplazar caracteres.
- Ejemplo:

python

```
s = "hello"
print(s.translate(str.maketrans("hl", "xy"))) # Output: "xxylo"
```

28. maxsplit

- Descripción: Establece el número máximo de divisiones para el método `split()`.
- Ejemplo:

python

```
s = "hello,world,python"
print(s.split(",", maxsplit=1)) # Output: ["hello",
"world,python"]
```

29. partition(sep)

- Descripción: Divide el string en tres partes según el separador `sep`.
- Ejemplo:

python

```
s = "hello:world:python"
print(s.partition(":")) # Output: ("hello", ":",
"world:python")
```

30. replace(old, new)

- Descripción: Reemplaza todas las apariciones del substring `old` con el substring `new`.
- Ejemplo:

python

```
s = "hello world"
print(s.replace("world", "python")) # Output: "hello python"
```

31. rfind(substring)

- Descripción: Devuelve la posición de la última aparición del substring `substring` en el string. Si no se encuentra, devuelve `-1`.
- Ejemplo:

python

```
s = "hello world"
print(s.rfind("world")) # Output: 6
```

32. rindex(substring)

- Descripción: Devuelve la posición de la última aparición del substring substring en el string. Si no se encuentra, lanza una excepción.
- Ejemplo:

python

```
s = "hello world"
print(s.rindex("world")) # Output: 6
```

33. rjust(width, fillchar)

- Descripción: Alinea el string a la derecha en un campo de ancho width y rellena con el carácter fillchar.
- Ejemplo:

python

```
s = "hello"
print(s.rjust(10, "*")) # Output: "*****hello"
```

34. rpartition(sep)

- Descripción: Divide el string en tres partes según el separador sep.
- Ejemplo:

python

```
s = "hello:world:python"
print(s.rpartition(":")) # Output: ("hello:world", ":",
"python")
```

35. rsplit(sep, maxsplit)

- Descripción: Divide el string en partes según el separador sep y devuelve una lista de las partes.
- Ejemplo:

python

```
s = "hello,world,python"
print(s.rsplit(",", maxsplit=1)) # Output: ["hello,world",
```



```
"python"]
```

36. `rstrip(chars)`

- Descripción: Elimina los caracteres de la cadena `chars` de la derecha del string.
- Ejemplo:

python

```
s = "  hello  "
print(s.rstrip()) # Output: "  hello"
```

37. `split(sep, maxsplit)`

- Descripción: Divide el string en partes según el separador `sep` y devuelve una lista de las partes.
- Ejemplo:

python

```
s = "hello,world,python"
print(s.split(",", maxsplit=1)) # Output: ["hello",
"world,python"]
```

38. `splitlines()`

- Descripción: Divide el string en líneas y devuelve una lista de las líneas.
- Ejemplo:

python

```
s = "hello\nworld\npython"
print(s.splitlines()) # Output: ["hello", "world", "python"]
```

39. `startswith(prefix)`

- Descripción: Verifica si el string comienza con el prefijo `prefix`.
- Ejemplo:

python

```
s = "hello world"
print(s.startswith("hello")) # Output: True
```

40. strip(chars)

- Descripción: Elimina los caracteres de la cadena `chars` de la izquierda y derecha del string.
- Ejemplo:

python

```
s = "  hello  "
print(s.strip()) # Output: "hello"
```

41. swapcase()

- Descripción: Intercambia las letras mayúsculas y minúsculas en el string.
- Ejemplo:

python

```
s = "Hello World"
print(s.swapcase()) # Output: "hELLO wORLD"
```

42. title()

- Descripción: Convierte el string a título (primera letra de cada palabra en mayúscula).
- Ejemplo:

python

```
s = "hello world"
print(s.title()) # Output: "Hello World"
```

43. translate(table)

- Descripción: Reemplaza caracteres según el diccionario de traducción `table`.
- Ejemplo:

python

```
s = "hello"
print(s.translate(str.maketrans("hl", "xy"))) # Output: "xxylo"
```

44. upper()

- Descripción: Convierte el string a mayúsculas.
- Ejemplo:

python

```
s = "hello"
print(s.upper()) # Output: "HELLO"
```

45. zfill(width)

- Descripción: Alinea el string a la izquierda en un campo de ancho width y rellena con ceros.
- Ejemplo:

python

```
s = "123"
print(s.zfill(5)) # Output: "00123"
```

46. encode(encoding, errors)

- Descripción: Codifica el string según el esquema de codificación encoding y maneja errores según errors.
- Ejemplo:

python

```
s = "hello"
print(s.encode("utf-8", "replace")) # Output: b'hello'
```

47. decode(encoding, errors)

- Descripción: Decodifica el string según el esquema de codificación encoding y maneja errores según errors.
- Ejemplo:

python

```
s = b"hello"
print(s.decode("utf-8", "replace")) # Output: "hello"
```

48. format_map(mapping)

- Descripción: Formatea el string según el diccionario mapping.
- Ejemplo:

python

```
s = "Hello, {name}!"
print(s.format_map({"name": "world"})) # Output: "Hello,
```

```
world!"
```

49. format(*args, **kwargs)

- Descripción: Formatea el string según los argumentos args y kwargs.
- Ejemplo:

python

```
s = "Hello, {}!"  
print(s.format("world")) # Output: "Hello, world!"
```

50. format_spec()

- Descripción: Devuelve la especificación de formato para el string.
- Ejemplo:

python

```
s = "hello"  
print(s.format_spec()) # Output: ""
```

51. index(substring)

- Descripción: Devuelve la posición de la primera aparición del substring substring en el string. Si no se encuentra, lanza una excepción.
- Ejemplo:

python

```
s = "hello world"  
print(s.index("world")) # Output: 6
```

52. isalnum()

- Descripción: Verifica si el string solo contiene caracteres alfanuméricos.
- Ejemplo:

python

```
s = "hello123"  
print(s.isalnum()) # Output: True
```

53. isalpha()

- Descripción: Verifica si el string solo contiene caracteres alfabéticos.

- Ejemplo:

python

```
s = "hello"  
print(s.isalpha()) # Output: True
```

54. isdecimal()

- Descripción: Verifica si el string solo contiene caracteres decimales.
- Ejemplo:

python

```
s = "123"  
print(s.isdecimal()) # Output: True
```

55. isdigit()

- Descripción: Verifica si el string solo contiene caracteres numéricos.
- Ejemplo:

python

```
s = "123"  
print(s.isdigit()) # Output: True
```